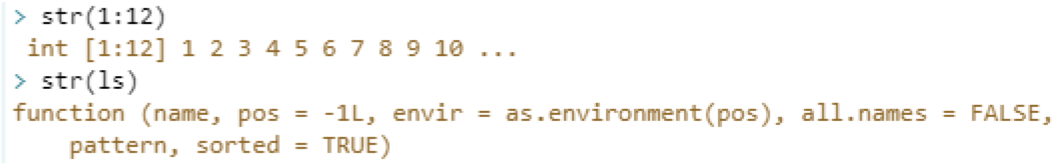
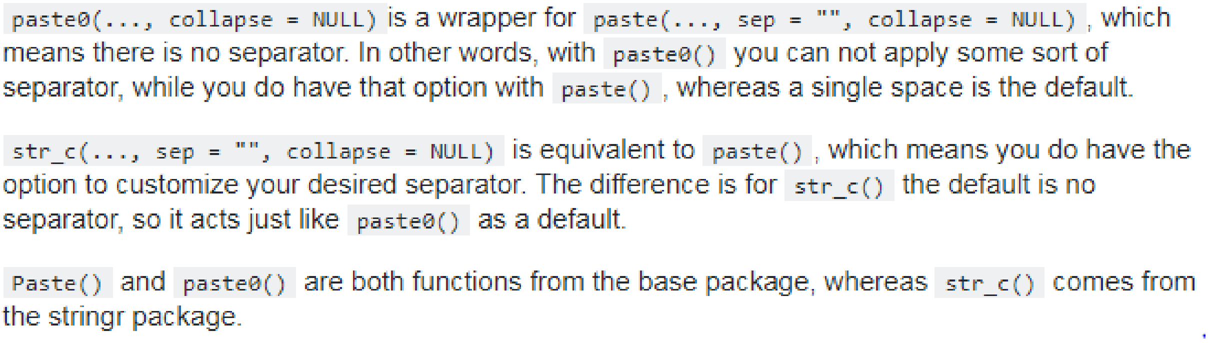
3. What is the major difference between str() and paste() show an example.

Str() will give the structure of the data set.



```
> str(1:12)
 int [1:12] 1 2 3 4 5 6 7 8 9 10 ...
> str(ls)
function (name, pos = -1L, envir = as.environment(pos), all.names = FALSE,
    pattern, sorted = TRUE)
```



`paste0(..., collapse = NULL)` is a wrapper for `paste(..., sep = "", collapse = NULL)`, which means there is no separator. In other words, with `paste0()` you can not apply some sort of separator, while you do have that option with `paste()`, whereas a single space is the default.

`str_c(..., sep = "", collapse = NULL)` is equivalent to `paste()`, which means you do have the option to customize your desired separator. The difference is for `str_c()` the default is no separator, so it acts just like `paste0()` as a default.

`Paste()` and `paste0()` are both functions from the base package, whereas `str_c()` comes from the stringr package.